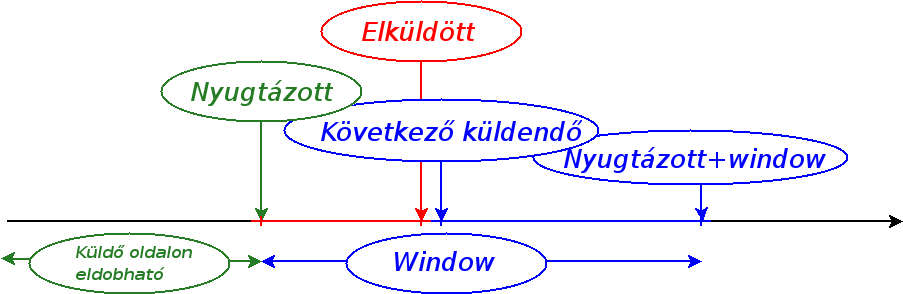
15. tétel - TCP flow control

# Sliding window



* adatáramlás szabályozása
* window ~ buffer mérete
* a csomagküldés telejsen hibamentes és tökéletes: állandó window
* nyílik: jobbra megy
* zárás: 0 lesz a mérete
* Shrink: jobb széle balra megy

# Kapcsolatépítés

* TCPkapcsolat építés 3 lépése:
  + kezdeményező SYN-t küld
  + Jön egy ACK és egy SYN
  + Visszamegy az ACK
* Kezdeményező: aktív open
* elszenvedett: passzív open

# Kapcsolat zárás

* kezdeményező küld egy FIN-t B-nek
* kap egy ACK-ot
* ha már B sem akar küldeni visszaküld egy FIN-t
* és a kezdeményező küld egy ACK-ot
* ezután megszánik a kapcsolat.
* Kezdeményező: aktív close
* Elszenvedett: passzív close

# TCP syn flood támadás

* + egy gép folyton SYN flages csomagokat küld random portokról/IP-kről
  + a támadó gép küldené az ACKokat, de nem bír velük -> DOS

# TIME\_WAIT

* Az aktív closet végrehajtó utolsó állapotánál
* már megkapta a FIN-t, és elküldte az ACK-ot
* de mi van ha az ACK elveszik, ezért vár 2MSL ideig utána zár
* MSL: ennyi ideig lehet egy szegmens a hálózaon

# TCP reset

* A és B kapcsolatban van
* E bejuttatt egy Resetet és így felbontja a kapcsolatot
* IP-ket portokat, sequence numbert tudni kell hozzá

# TCP retransmission timer (RTO)

* Ha RTO ideig nem jön ACK egy csomagra, újra kell küldeni, mert elveszett
* egy időérték - Round Time Tripből (idő amíg odaér + ACK ideje) számoljuk

# Delayed ack

* Nem azonnal küldünk ACK-ot
* várunk még, hátha jön még adat
* vagy hátha mi küldünk, amivel tudunk egyben nyugtázni is (piggy back)

# Slow start

* nem csak a fogadó, hanem a küldő is flow controlt alkalmaz
* kiegyensúlyozza a hálózat sebességét
* lassan növeljük a csomagokat amiket küdlünk, a hálózat max kapacitásáig
* congestion window - reciver window - egyensúly

# Passzív ftp

* Az adatkapcsolatot nem a szerver nyitja hanem a kliens
* PASV paranccsal
* kliens ephemeral portja és a kapott port között nyílik kapcsoal